UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/823,155	03/29/2001	Daniel R. Gaur	42390P10774	8114

8791 7590 01/07/2004

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD, SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER	
BROSS, EDWARD J	

ART UNIT	PAPER NUMBER
2126	5

DATE MAILED: 01/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/823,155	GAUR ET AL.	
	Examiner	Art Unit	
	Edward Bross	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☒ Claim(s) *7, 11, 23, 26, 30* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Specification

Content of Specification

(a) Title of the Invention: See 37 CFR 1.72(a) and MPEP § 606. The title of the invention should be placed at the top of the first page of the specification unless the title is provided in an application data sheet. The title of the invention should be brief but technically accurate and descriptive, preferably from two to seven words may not contain more than 500 characters.

(b) Cross-References to Related Applications: See 37 CFR 1.78 and MPEP § 201.11.

(c) Statement Regarding Federally Sponsored Research and Development: See MPEP § 310.

(d) Incorporation-By-Reference Of Material Submitted On a Compact Disc: The specification is required to include an incorporation-by-reference of electronic documents that are to become part of the permanent United States Patent and Trademark Office records in the file of a patent application. See 37 CFR 1.52(e) and MPEP § 608.05. Computer program listings (37 CFR 1.96(c)), "Sequence Listings" (37 CFR 1.821(c)), and tables having more than 50 pages of text were permitted as electronic documents on compact discs beginning on September 8, 2000.

Or alternatively, Reference to a "Microfiche Appendix": See MPEP § 608.05(a). "Microfiche Appendices" were accepted by the Office until March 1, 2001.

(e) Background of the Invention: See MPEP § 608.01(c). The specification should set forth the Background of the Invention in two parts:

(1) Field of the Invention: A statement of the field of art to which the invention pertains. This statement may include a paraphrasing of the applicable U.S. patent classification definitions of the subject matter of the claimed invention. This item may also be titled "Technical Field."

(2) Description of the Related Art including information disclosed under 37 CFR 1.97 and 37 CFR 1.98: A description of the related art known to the applicant and including, if applicable, references to specific related art and problems involved in the prior art which are solved by the applicant's invention. This item may also be titled "Background Art."

(f) Brief Summary of the Invention: See MPEP § 608.01(d). A brief summary or general statement of the invention as set forth in 37 CFR 1.73. The summary is separate and distinct from the abstract and is directed toward the invention rather than the disclosure as a whole. The summary may point out the advantages of the invention or how it solves problems previously existent in the prior art (and preferably indicated in the Background of the Invention). In chemical cases it should point out in general terms the utility of the invention. If possible, the nature and gist of the invention or the inventive concept should be set forth. Objects of the invention should be treated briefly and only to the extent that they contribute to an understanding of the invention.

(g) Brief Description of the Several Views of the Drawing(s): See MPEP § 608.01(f). A reference to and brief description of the drawing(s) as set forth in 37 CFR 1.74.

(h) Detailed Description of the Invention: See MPEP § 608.01(g). A description of the preferred embodiment(s) of the invention as required in 37 CFR 1.71. The description should be as short and specific as is necessary to describe the invention adequately and accurately. Where elements or groups of elements, compounds, and processes, which are conventional and generally widely known in the field of the invention described and their exact nature or type is not necessary for an understanding and use of the invention by a person skilled in the art, they should not be described in detail. However, where particularly complicated subject matter is involved or where the elements, compounds, or processes may not be commonly or widely known in the field, the specification should refer to another patent or readily available publication which adequately describes the subject matter.

(i) Claim or Claims: See 37 CFR 1.75 and MPEP § 608.01(m). The claim or claims must commence on separate sheet or electronic page (37 CFR 1.52(b)(3)). Where a claim sets forth a plurality of elements or steps, each element or step of the claim should be separated by a line indentation. There may be plural indentations to further segregate subcombinations or related steps. See 37 CFR 1.75 and MPEP § 608.01(i)-(p).

(j) Abstract of the Disclosure: See MPEP § 608.01(f). A brief narrative of the disclosure as a whole in a single paragraph of 150 words or less commencing on a separate sheet following the claims. In an international application which has entered the national stage (37 CFR 1.491(b)), the applicant need not submit an abstract commencing on a separate sheet if an abstract was published with the international application under PCT Article 21. The abstract that appears on the cover page of the pamphlet published by the International Bureau (IB) of the World Intellectual Property Organization (WIPO) is the abstract that will be used by the USPTO. See MPEP § 1893.03(e).

(k) Sequence Listing, See 37 CFR 1.821-1.825 and MPEP §§ 2421-2431. The requirement for a sequence listing applies to all sequences disclosed in a given application, whether the sequences are claimed or not. See MPEP § 2421.02.

1. The disclosure is objected to because of the following informalities: There is no "Summary of the Invention" heading.

Appropriate correction is required.

Claim Objections

1. Claim 7 is objected to because of the following informalities: the phrase "with one of the first deferred procedure call" should read "with one of the first deferred procedure calls". Appropriate correction is required.

2. Claim 11 is objected to because of the following informalities: the phrase "with one of the first deferred procedure call" should read "with one of the first deferred procedure calls". Appropriate correction is required.

3. Claim 23 is objected to because of the following informalities: the phrase "the at least one other deferred procedure call on the second thread thread" should read "the at least one other deferred procedure call on the second thread". Appropriate correction is required.

4. Claim 26 is objected to because of the following informalities: the phrase "with one of the first deferred procedure call" should read "with one of the first deferred procedure calls". Appropriate correction is required.

5. Claim 30 is objected to because of the following informalities: the phrase "with one of the first deferred procedure call" should read "with one of the first deferred procedure calls". Appropriate correction is required.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-30 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Linux (version 2.3.99-pre3 March 23, 2000) described in Linux Device Drivers (Rubini and Corbet).

3. As to claim 1, Linux discloses a method comprising:

requesting a first deferred procedure call for a first interrupt event (e.g. request_irq(unsigned int irq, void (*handler)(int, void *, struct pt_regs *), unsigned long flags, const char *dev_name, void *dev_id) as defined in arch/i386/kernel/irq.c and used in drivers/char/pc_keyb.c where 'handler' is the deferred procedure call);

requesting at least one other deferred procedure call for a second interrupt event (e.g. the same function as above as used in drivers/net/sis900.c);

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource (e.g. request_irq(...) above "...allocates interrupt resources..." arch/i386/kernel/irq.c line 640);

processing the first interrupt event with the first deferred procedure call (e.g. the deferred procedure will be called when the corresponding interrupts occur arch/i386/kernel/irq.c lines 439-443); and

processing the second interrupt event with at least one other deferred procedure call (e.g. same as above when the second interrupt event occurs after the first interrupt event).

4. As to claim 2, Linux discloses the limitations of claim 1 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a single thread of execution (e.g. when running on a single processor system, since as implied in Rubini and Corbet by endnote 38 on page 36, the kernel is not preemptable, there is only a single kernel thread of execution outside of interrupt handlers. If only fast handlers are used, interrupt reporting will be disabled, ensuring that both handlers will run sequentially on the same thread, p.11 "Fast and Slow Handlers"); and

executing the first deferred procedure call and the at least one other deferred procedure call on the single thread (e.g. as above, these procedures will be called on the single thread).

5. As to claim 3, Linux discloses the limitations of claim 1 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a plurality of threads (e.g. the top half routine, the bottom half routines, and other user processes running on the processor Rubini and Corbet p. 17 "Tasklets and Bottom-Half Processing"); and

executing the first deferred procedure call on one thread of the plurality of threads while executing the at least one other deferred procedure call on another thread of the plurality of threads (e.g. when processing an interrupt in a bottom half routine and

a second interrupt occurs, processing of the second interrupt will be done by a top half routine in a separate thread from the currently executing bottom half, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing").

6. As to claim 4, Linux discloses the limitations of claim 1 as above and:

assigning the first deferred procedure call to a resource comprising a first thread of a processor (e.g. during top half processing of an interrupt, a bottom half is scheduled, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing");

assigning the at least one other deferred procedure call to a resource comprising a second thread of the processor (e.g. during top half processing of a second interrupt, another bottom half is scheduled, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing"); and

executing the first deferred procedure call on the first thread while executing the at least one other deferred procedure call on the second thread. (e.g. as described above, the two handlers will each process their respective interrupts in separate threads)

7. As to claim 5, Linux discloses the limitations of claim 1 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a multi-processor system (e.g. Rubini and Corbet p. 18, third paragraph "SMP systems"); and

executing the first deferred procedure call on one processor of the multi-processor system while executing the at least one other deferred procedure call on another processor of the multi-processor system (e.g. Rubini and Corbet p. 18, third paragraph

"Tasklets can run in parallel with other tasklets on SMP systems." where tasklets are a type of bottom half handler).

8. As to claim 6, Linux discloses the limitations of claim 1 as above and:

assigning the first deferred procedure call to a resource comprising a first processor;

assigning the at least one other deferred procedure call to a resource comprising a second processor (e.g. Rubini and Corbet p. 18 "Tasklets are also guaranteed to run on the same CPU as the function that first schedules them"); and

executing the first deferred procedure call on the first processor while executing the at least one other deferred procedure call on the second processor (e.g. as above, when the tasklets are executed one will be on the first processor, the other on the second processor).

9. As to claim 7, Linux discloses the limitations of claim 1 as above and:

processing another interrupt event with one of the first deferred procedure calls and the at least one other deferred procedure call (e.g. this will occur implicitly whenever the first interrupt occurs again).

10. As to claim 8, Linux discloses a method comprising:

requesting a first deferred procedure call for a first interrupt event (e.g. request_irq(unsigned int irq, void (*handler)(int, void *, struct pt_regs *), unsigned long flags, const char *dev_name, void *dev_id) as defined in arch/i386/kernel/irq.c and used in drivers/char/pc_keyb.c line 744 where 'handler' is the deferred procedure call);

requesting at least one other deferred procedure call for a second interrupt event (e.g. the

same function as above as used in drivers/net/sis900.c line 476); and

processing the first interrupt event with the first deferred procedure call (e.g. the deferred procedure will be called when the corresponding interrupts occur arch/i386/kernel/irq.c lines 439-443) while processing the second interrupt event with the at least one other deferred procedure call (e.g. same as above when the second interrupt event occurs after the first interrupt event).

11. As to claim 9, Linux discloses the limitations of claim 8 as above and:

executing the first deferred procedure call on a first thread of a processor; and

executing the at least one other deferred procedure call on a second thread of the processor (e.g. as described in the above 102 rejection of claim 4, the two handlers will each process their respective interrupts in separate threads).

12. As to claim 10, Linux discloses the limitations of claim 8 as above and:

executing the first deferred procedure call on a first processor; and

executing the at least one other deferred procedure call on a second processor.

(e.g. Rubini and Corbet p. 18 "Tasklets are also guaranteed to run on the same CPU as the function that first schedules them" implying that when two interrupts are delivered to two different CPUs, the tasklets, or deferred procedures, they will each run on separate processor).

13. As to claim 11, Linux discloses the limitations of claim 8 as above and:

processing another interrupt event with one of the first deferred procedure calls and the at least one other deferred procedure call (as stated in the above 102 rejection of claim 7, e.g. this will occur implicitly whenever the first interrupt occurs again).

14. As to claim 12, Linux discloses a driver comprising:

an interrupt handler to identify interrupt events (e.g. arch/i386/kernel/irq.c lines 427-451 handle_IRQ_event(...)); and

at least two deferred procedure calls, each of the at least two deferred procedure calls to process at least one of the interrupt events (e.g. drivers/char/pc_keyb.c keyboard_interrupt(...) lines 469-479 and drivers/net/sis900.c sis900_interrupt(...) lines 850-897).

15. As to claim 13, Linux discloses the limitations of claim 12 as above and:

the interrupt handler to assign the at least two deferred procedure calls to a resource for execution (e.g. drivers/char/pc_keyb.c line 744 and drivers/net/sis900.c line 476).

16. As to claim 14, Linux discloses the limitations of claim 12 as above and:

the interrupt handler to assign on of the at least two deferred procedure calls to a first resource for execution and another of the at least two deferred procedure calls to a second resource for execution (e.g. occurs when the two interrupts are handled by two different processors where the deferred procedures will be executed on the different processors. The use of spinlocks in the keyboard_interrupt(...) and sis900_interrupt(...) implies that these functions may run in parallel on different processors in SMP systems).

17. As to claim 15, Linux discloses a computer system comprising:

a driver stored in a memory of the computer system, the driver including

an interrupt handler to identify interrupt events (e.g. arch/i386/kernel/irq.c lines 427-451 handle_IRQ_event(...)); and

at least two deferred procedure calls, each of the at least two deferred procedure

calls to process at least one of the interrupt events (e.g. drivers/char/pc_keyb.c keyboard_interrupt(...) lines 469-479 and drivers/net/sis900.c sis900_interrupt(...) lines 850-897), and

a processor to execute the at least two deferred procedure calls (e.g. the implied processor on which the Linux kernel is run).

18. As to claim 16, Linux discloses the limitations of claim 15 as above and:

the interrupt handler to assign the at least two deferred procedure calls to a single thread exhibited by the processor for execution (e.g. when running on a single processor system, since as implied in Rubini and Corbet by endnote 38 on page 36, the kernel is not preemptable, there is only a single kernel thread of execution outside of interrupt handlers. If only fast handlers are used, interrupt reporting will be disabled, ensuring that both handlers will run sequentially on the same thread, p.11 "Fast and Slow Handlers").

19. As to claim 17, Linux discloses the limitations of claim 15 as above and:

the interrupt handler to assign a first of the at least two deferred procedure calls to one thread of the processor and another of the at least two deferred procedure calls to a second thread of the processor for execution (e.g. col. 14 line 65 – col. 15 line 19).

20. As to claim 18, Linux discloses the limitations of claim 15 as above and:

the interrupt handler to assign one of the at least two deferred procedure calls to the processor and another of the at least two deferred procedure calls to a second processor (e.g. Rubini and Corbet p. 18 "Tasklets are also guaranteed to run on the same CPU as the function that first schedules them" implying that when two interrupts are delivered to two different CPUs, the tasklets, or deferred procedures, they will each run on separate processor).

21. As to claim 19, Linux discloses the limitations of claim 15 as above and:

at least one peripheral device, the interrupt events associated with the at least one peripheral device (e.g. keyboard_interrupt(...) associated with the keyboard).

22. As to claim 20, Linux discloses an article of manufacture comprising:

a machine accessible medium, the machine accessible medium providing instructions that, when executed by a machine, cause a machine to:

request a first deferred procedure call for a first interrupt event (e.g. request_irq(unsigned int irq, void (*handler)(int, void *, struct pt_regs *), unsigned long flags, const char *dev_name, void *dev_id) as defined in arch/i386/kernel/irq.c and used in drivers/char/pc_keyb.c where 'handler' is the deferred procedure call);

request at least one other deferred procedure call for a second interrupt event (e.g. the same function as above as used in drivers/net/sis900.c);

assign he first deferred procedure call and the at least one other deferred procedure call to a resource (e.g. request_irq(...) above "...allocates interrupt resources..." arch/i386/kernel/irq.c line 640);

process the first interrupt event with the first deferred procedure call (e.g. the deferred procedure will be called when the corresponding interrupts occur arch/i386/kernel/irq.c lines 439-443); and

process the second interrupt event with the at least one other deferred procedure call (e.g. same as above when the second interrupt event occurs after the first interrupt event).

23. As to claim 21, Linux discloses the limitations of claim 20 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a single thread of execution (e.g. when running on a single processor system, since as implied in Rubini and Corbet by endnote 38 on page 36, the kernel is not preemptable, there is only a single kernel thread of execution outside of interrupt handlers. If only fast handlers are used, interrupt reporting will be disabled, ensuring that both handlers will run sequentially on the same thread, p.11 "Fast and Slow Handlers"); and

execute the first deferred procedure call and the at least one other deferred procedure call on the single thread (e.g. as above, these procedures will be called on the single thread).

24. As to claim 22, Linux discloses the limitations of claim 20 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a plurality of threads (e.g. the top half routine, the bottom half routines, and other user processes running on the processor Rubini and Corbet p. 17 "Tasklets and Bottom-Half Processing"); and

execute the first deferred procedure call on one thread of the plurality of threads while executing the at least one other deferred procedure call on another thread of the plurality of threads (e.g. when processing an interrupt in a bottom half routine and a second interrupt occurs, processing of the second interrupt will be done by a top half routine in a separate thread from the currently executing bottom half, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing").

25. As to claim 23, Linux discloses the limitations of claim 20 as above and:

assigning the first deferred procedure call to a resource comprising a first thread of a processor (e.g. during top half processing of an interrupt, a bottom half is scheduled, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing");

assigning the at least one other deferred procedure call to a resource comprising a second thread of the processor (e.g. during top half processing of a second interrupt, another bottom half is scheduled, Rubini and Corbet p 17 "Tasklets and Bottom-Half Processing"); and

executing the first deferred procedure call on the first thread while executing the at least one other deferred procedure call on the second thread. (e.g. as described above, the two handlers will each process their respective interrupts in separate threads)

26. As to claim 24, Linux discloses the limitations of claim 20 as above and:

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a multi-processor system (e.g. Rubini and Corbet p. 18, third paragraph "SMP systems"); and

executing the first deferred procedure call on one processor of the multi-processor system while executing the at least one other deferred procedure call on another processor of the multi-processor system (e.g. Rubini and Corbet p. 18, third paragraph "Tasklets can run in parallel with other tasklets on SMP systems." where tasklets are a type of bottom half handler).

27. As to claim 25, Linux discloses the limitations of claim 20 as above and:

assigning the first deferred procedure call to a resource comprising a first processor;

assigning the at least one other deferred procedure call to a resource comprising a second

processor (e.g. Rubini and Corbet p. 18 "Tasklets are also guaranteed to run on the same CPU as the function that first schedules them"); and

executing the first deferred procedure call on the first processor while executing the at least one other deferred procedure call on the second processor (e.g. as above, when the tasklets are executed one will be on the first processor, the other on the second processor).

28. As to claim 26, Linux discloses the limitations of claim 20 as above and:

processing another interrupt event with one of the first deferred procedure calls and the at least one other deferred procedure call (e.g. this will occur implicitly whenever the first interrupt occurs again).

29. As to claim 27, Linux discloses an article of manufacture comprising:

requesting a first deferred procedure call for a first interrupt event (e.g. request_irq(unsigned int irq, void (*handler)(int, void *, struct pt_regs *), unsigned long flags, const char *dev_name, void *dev_id) as defined in arch/i386/kernel/irq.c and used in drivers/char/pc_keyb.c line 744 where 'handler' is the deferred procedure call);

requesting at least one other deferred procedure call for a second interrupt event (e.g. the same function as above as used in drivers/net/sis900.c line 476); and

processing the first interrupt event with the first deferred procedure call (e.g. the deferred procedure will be called when the corresponding interrupts occur arch/i386/kernel/irq.c lines 439-443) while processing the second interrupt event

with the at least one other deferred procedure call (e.g. same as above when the second interrupt event occurs after the first interrupt event).

30. As to claim 28, Linux discloses the limitations of claim 27 as above and:

executing the first deferred procedure call on a first thread of a processor; and

executing the at least one other deferred procedure call on a second thread of the processor (e.g. as described in the above 102 rejection of claim 4, the two handlers will each process their respective interrupts in separate threads).

31. As to claim 29, Linux discloses the limitations of claim 27 as above and:

executing the first deferred procedure call on a first processor; and

executing the at least one other deferred procedure call on a second processor.

(e.g. Rubini and Corbet p. 18 "Tasklets are also guaranteed to run on the same CPU as the function that first schedules them" implying that when two interrupts are delivered to two different CPUs, the tasklets, or deferred procedures, they will each run on separate processor).

32. As to claim 30, Kleiman discloses the limitations of claim 27 as above and:

processing another interrupt event with one of the first deferred procedure calls and the at least one other deferred procedure call (as stated in the above 102 rejection of claim 7, e.g. this will occur implicitly whenever the first interrupt occurs again).

Conclusion

33. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See U.S. Patent #5,515,538 Kleiman, U.S. Patent #5,179,702 Spix et al. col. 10 lines 32-51 and U.S. Patent #5,911,078 Anderson col. 4 line 25 – col. 5 line 12 and col. 9 lines 7-11.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Edward Bross whose telephone number is 305-8754. The examiner can normally be reached on Mon-Fri 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on 305-8498. The fax phone number for the organization where this application or proceeding is assigned is 308-5355.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

EB

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100